

Medición y comparación de prestaciones

- Problemas

- Considérese un sistema informático (con palabras de 32 bits) en el que la activación de un monitor software implica la ejecución de 1200 instrucciones máquina. Si el procesador del sistema tiene una velocidad de ejecución de 200 MIPS, se pide:
 - Calcular el valor que ha de tener un periodo de muestreo si se quiere una sobrecarga (*overhead*) del 5%.
 - Con el periodo de muestreo obtenido, calcular el tamaño del fichero de resultados generado tras un periodo de medida de 4 horas si en cada activación se graban 5 palabras.
 - Cual es la sobrecarga si cada operación de almacenamiento de datos se ejecuta en 0,28s.
 - Cual es la sobrecarga si se emplea un buffer de 32K palabras y cada operación de almacenamiento en el buffer se realiza en 0,01s.
 - ¿Qué ocurre si se toma en cuenta la velocidad de volcado de datos a disco desde el buffer? ¿Cuál sería el resultado si las operaciones de volcado de datos detienen el sistema durante 0,5 segundos?

- El resultado de la ejecución de la orden `vmstat 2 16` en un sistema informático que utiliza Linux como sistema operativo es el que se presenta en la siguiente transparencia.

- ¿Cuál es la duración del periodo de medida?
- ¿Tienen algún interés los datos ofrecidos en la primera línea? Explicar razonadamente la respuesta.
- ¿Cuál es el número medio de procesos que están esperando a ser ejecutados?
- ¿Cuál es la utilización media del procesador en modo usuario?
- Calcúlese la sobrecarga del procesador producida por el sistema operativo.
- ¿Ha habido actividad de intercambio de memoria (*swapping*) durante el periodo de medida? ¿Por qué?
- ¿Cuál ha sido la actividad con los dispositivos de bloques?

vmstat 2 16

procs				memory		swap		io		system		cpu			
r	b	w	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id
0	0	0	0	26324	164928	19672	0	0	0	1	1	13	5	11	1
1	0	0	0	26324	164928	19672	0	0	0	0	113	97	12	29	59
5	0	0	0	26204	164928	19724	0	0	18	0	159	93	51	6	43
2	0	0	0	26136	164928	19728	0	0	4	0	118	127	68	0	32
2	0	0	0	26136	164928	19728	0	0	0	7	166	125	99	1	0
2	0	0	0	26136	164928	19728	0	0	0	0	121	160	99	1	0
2	0	0	0	26136	164928	19728	0	0	5	0	110	114	99	1	0
4	0	0	0	25604	164928	20048	0	0	2	0	440	231	81	19	0
2	0	0	0	25592	164928	20048	0	0	1	0	108	136	91	10	0
2	0	0	0	25792	164928	20048	0	0	0	1	120	182	98	2	0
3	0	0	0	25792	164928	20048	0	0	0	0	104	108	99	1	0
6	0	0	0	25792	164928	20048	0	0	32	0	115	137	97	3	0
8	0	0	0	25732	164928	20052	0	0	2	6	156	133	96	4	0
3	0	0	0	25732	164928	20052	0	0	0	0	103	81	78	22	0
3	0	0	0	25732	164928	20052	0	0	0	4	134	80	79	21	0
3	0	0	0	25732	164928	20052	0	0	0	0	111	100	76	24	0

- Tras ejecutar la orden **top** en un sistema informático monoprocesador que utiliza Linux como sistema operativo se ha obtenido la información de actividad que se muestra en la siguiente transparencia.

- ¿Cuánta memoria física tiene este computador?
- ¿Qué porcentaje de la memoria física está usada actualmente?
- ¿Cuál es la utilización media del procesador?
- ¿Cuál es la carga media a lo largo de los últimos 15 minutos?
- ¿Cómo es la evolución de la carga medida del sistema, ascendente o descendente?
- ¿Hay algún proceso en ejecución con baja prioridad? ¿Cuál?
- ¿Hay algún proceso residente en el disco debido al intercambio de memoria (*swapped out*)? ¿Cuál?
- ¿Cuánta memoria física está ocupando el programa **trilog**?
- ¿Cuánto tiempo lleva en ejecución el programa **sieve**?

Top

```
$ top
1:27pm up 1 day, 1:11, 3 users, load average 2.46, 0.80, 0.28
53 processes: 48 sleeping, 5 running, 0 zombie, 0 stopped
CPU states: 82.5% user, 0.5% system, 17.0% nice, 0.0% idle
Mem: 256464K av, 251672K used, 4792K free, 0K shrd, 22792K buff
Swap: 136512K av, 1956K used, 134556K free, 67484K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LC	%CPU	%MEM	TIME	COMMAND
6221	pau	0	0	600	600	484	R	0	27.5	0.2	0:43	sieve
6223	pau	0	0	596	596	480	R	0	27.3	0.2	0:28	qnap
6224	pau	0	0	596	596	480	R	0	27.3	0.2	0:26	qnap
6230	pau	0	10	584	584	468	R N	0	17.0	0.2	1:64	trilog
6231	pau	19	0	980	980	776	R	0	0.3	0.3	0:00	top
2715	root	20	0	23176	6212	2064	S	0	0.1	2.4	0:01	X
1	root	20	0	84	64	44	S	0	0.0	0.0	0:03	init
2	root	20	0	0	0	0	SW	0	0.0	0.0	0:00	keventd
3	root	20	0	0	0	0	SW	0	0.0	0.0	0:00	kpmc
4	root	20	19	0	0	0	SWN	0	0.0	0.0	0:00	ksoftiq
5	root	20	0	0	0	0	SW	0	0.0	0.0	0:11	kswapd
6	root	2	0	0	0	0	SW	0	0.0	0.0	0:00	kbdflush
7	root	20	0	0	0	0	SW	0	0.0	0.0	0:00	kupdatd
8	root	19	0	0	0	0	SW	0	0.0	0.0	0:01	kinoded
366	root	20	0	600	596	488	S	0	0.0	0.0	0:02	syslogd
369	root	20	0	972	964	360	S	0	0.0	0.0	0:03	klogd
405	root	20	0	0	0	0	SW	0	0.0	0.0	0:00	khud
487	root	20	0	332	316	244	S	0	0.0	0.1	0:00	portmap
530	root	20	0	340	312	252	S	0	0.0	0.1	0:00	atd
554	root	14	0	900	816	684	S	0	0.0	0.3	0:02	sshd

-
- Explicar las diferencias entre las siguientes ordenes:

sadc 3 10 datos_sadc

sar 3 10 > datos_sar

-
- Un programa de cálculo numérico se ha instrumentado mediante la herramienta **gprof** y el resultado obtenido después de ejecutarlo ha sido el que se presenta en la siguiente transparencia.
 - Teniendo en cuenta esos datos se pide:
 - Indicar el numero de muestras utilizada para generar los resultados.
 - ¿Cuál es el procedimiento más rápido? ¿Cuál es el más lento? Justificar las respuestas.
 - Si el procedimiento más lento de los cuatro se substituye por una nueva versión cuatro veces más rápida, ¿cuánto tiempo tardará en ejecutarse el programa? ¿Cuál es la mejora de rendimiento conseguida sobre el programa original con esta mejora?
 - Si los procedimientos **raíz** y **multi** se substituyen por nevas versiones tres y dos veces más rápidas, respectivamente, ¿qué mejora se obtendrá en el tiempo de ejecución del programa original?
 - ¿Cuánto se debería mejorar el procedimiento **raíz** si se quiere que el programa original se ejecute en menos de 250 segundos?

Flat profile:

Each sample counts as 0.02 seconds.

%	cumulative	self		self	total	
time	seconds	seconds	calls	ms/call	ms/call	name
51.44	170.52	170.52	12	14210.00	14210.00	raíz
26.34	257.83	87.31	2	43655.00	43655.00	multi
20.79	326.74	68.91	32	2153.44	2153.44	tangente
1.43	331.50	4.76	87	54.71	54.71	suma

-
- Un sistema informático que trabaja con el sistema operativo Linux tiene instalado el monitor de actividad **sar** (*system activity reporter*). Este monitor se activa cada 20 minutos y tarda 420 milisegundos en ejecutarse por cada activación. En cada una de las activaciones se recoge información del sistema, se construye un registro de datos con esta información, y se añade al fichero histórico **saDD** de día **DD** correspondiente.

- Calcular la sobrecarga (*overhead*) que genera este programa sobre el sistema informático.
- Determinar el tamaño del directorio **/var/log/sa** a lo largo de dos semanas si el registro de datos generado por cada activación ocupa 3 KB.
- Si el volumen máximo del directorio **/var/log/sa** es 150 MB, ¿cuántos ficheros históricos **saDD** se pueden almacenar?
- Describir que efecto tiene la orden **sar -d 2 30**.
- Indicar que información histórica mostrará la siguiente orden:
sar -A -s 9:00:00 -e 12:00:00 -f /var/log/sa/sa01